CLAIMS

What is claimed is:

1. A method comprising:

determining a processor state of a processor upon expiration of a system management interrupt (SMI) timer, the processor state being one of an operational state and a low power state;

loading the SMI timer with a timer value based on the processor state, the timer value being one of a first value and a second value; and

transitioning the processor to one of the operational state and the low power state according to the processor state.

2. The method of claim 1 wherein loading the SMI timer comprises:

loading the SMI timer with the first value if the processor state is the operational state; and

loading the SMI timer with the second value if the processor state is the low power state.

3. The method of claim 2 wherein transitioning comprises:

transitioning the processor to the operational state if the processor state is the low power state; and

transitioning the processor to the reduced power state if the processor state is the operational state.

4. The method of claim 1 further comprising:

disabling the SMI timer if throttling is disabled; and

enabling the SMI timer if throttling is enabled.

5. The method of claim 1 further comprising:

generating a SMI access to a throttling state in response to an input/output (I/O) trap;

updating the throttling state if the access is a write; and

returning the throttling state if the access is a read.

6. The method of claim 5 wherein generating the SMI access comprises:

reporting the throttling state at a dummy address; and

generating the I/O trap by an SMI handler using the dummy address, the I/O trap generating the SMI access.

7. The method of claim 6 wherein generating the I/O trap comprises:

invoking the SMI handler to trap on the dummy address;

booting a power management OS;

loading the dummy address by the power management OS;

accessing the throttling state at the dummy address by the power management OS; and

generating the I/O trap by a chipset in response to accessing the throttling state by the power management OS.

8. The method of claim 4 wherein booting the power management OS comprises:

booting an Advanced Configuration and Power Interface (ACPI) OS.

9. The method of claim 3 wherein transitioning the processor to the low power state comprises:

transitioning the processor to one of a first power state, a second power state, a third power state, and a sleep state.

10. The method of claim 1 wherein loading the SMI timer comprises:

loading the SMI timer in a chipset.

11. A computer program product comprises:

a machine useable medium having computer program code embedded therein, the computer program product having:

computer readable program code to determine a processor state of a processor upon expiration of a system management interrupt (SMI) timer, the processor state being one of an operational state and a low power state;

computer readable program code to load the SMI timer with a timer value based on the processor state, the timer value being one of a first value and a second value; and

computer readable program code to transition the processor to one of the operational state and the low power state according to the processor state.

12. The computer program product of claim 11 wherein the computer readable program code to load the SMI timer comprises:

computer readable program code to load the SMI timer with the first value if the processor state is the operational state; and

computer readable program code to load the SMI timer with the second value if the processor state is the low power state.

13. The computer program product of claim 12 wherein the computer readable program code to transition comprises:

computer readable program code to transition the processor to the operational state if the processor state is the low power state; and

computer readable program code to transition the processor to the reduced power state if the processor state is the operational state.

14. The computer program product of claim 11 further comprising:

computer readable program code to disable the SMI timer if throttling is disabled; and

computer readable program code to enable the SMI timer if throttling is enabled.

15. The computer program product of claim 11 further comprising:

computer readable program code to generate a SMI access to a throttling state in response to an input/output (I/O) trap;

computer readable program code to update the throttling state if the access is a write; and

computer readable program code to return the throttling state if the access is a read.

16. The computer program product of claim 15 wherein the computer readable program code to generate the SMI access comprises:

computer readable program code to report the throttling state at a dummy address; and

computer readable program code to generate the I/O trap by an SMI handler using the dummy address, the I/O trap generating the SMI access.

17. The computer program product of claim 16 wherein the computer readable program code to generate the I/O trap comprises:

computer readable program code to invoke the SMI handler to trap on the dummy address;

computer readable program code to boot a power management OS;

computer readable program code to load the dummy address by the power management OS;

computer readable program code to access the throttling state at the dummy address by the power management OS; and

computer readable program code to generate the I/O trap by a chipset in response to accessing the throttling state by the power management OS.

18. The computer program product of claim 4 wherein the computer readable program code to boot the power management OS comprises:

computer readable program code to boot an Advanced Configuration and Power Interface (ACPI) OS.

19. The computer program product of claim 3 wherein the computer readable program code to transition the processor to the low power state comprises:

computer readable program code to transition the processor to one of a first power state, a second power state, a third power state, and a sleep state.

20. The computer program product of claim 11 wherein the computer readable program code to load the SMI timer comprises:

computer readable program code to load the SMI timer in a chipset.

21. A system comprising:

a processor;

a memory coupled to the processor to store a throttling emulator, the throttling emulator, when executed, causing the processor to:

determine a processor state of the processor upon expiration of a system management interrupt (SMI) timer, the processor state being one of an operational state and a low power state;

load the SMI timer with a timer value based on the processor state, the timer value being one of a first value and a second value; and

transition the processor to one of the operational state and the low power state according to the processor state.

22. The system of claim 21 wherein the throttling emulator causing the processor to load causes the processor to:

load the SMI timer with the first value if the processor state is the operational state; and

load the SMI timer with the second value if the processor state is the low power state.

23. The system of claim 22 wherein the throttling emulator causing the processor to transition causes the processor to:

transition the processor to the operational state if the processor state is the low power state; and

transition the processor to the reduced power state if the processor state is the operational state.

24. The system of claim 21 wherein the throttling emulator, when executed, further causes the processor to:

disable the SMI timer if throttling is disabled; and

enable the SMI timer if throttling is enabled.

25. The system of claim 21 wherein the throttling emulator further causes the processor to:

generate a SMI access to a throttling state in response to an input/output (I/O) trap;

update the throttling state if the access is a write; and

return the throttling state if the access is a read.

26. The system of claim 25 wherein the throttling emulator causing the processor to generate the SMI access causes the processor to:

report the throttling state at a dummy address; and

generate the I/O trap by an SMI handler using the dummy address, the I/O trap generating the SMI access.

27. The system of claim 26 wherein the throttling emulator causing the processor to generate the I/O trap causes the processor to:

invoke the SMI handler to trap on the dummy address;

boot a power management OS;

load the dummy address by the power management OS;

access the throttling state at the dummy address by the power management OS; and

generate the I/O trap by a chipset in response to accessing the throttling state by the power management OS.

28. The system of claim 24 wherein the throttling emulator causing the processor to boot the power management OS causes the processor to:

boot an Advanced Configuration and Power Interface (ACPI) OS.

29. The system of claim 23 wherein the throttling emulator causing the processor to transition the processor to the low power state causes the processor to:

transition the processor to one of a first power state, a second power state, a third power state, and a sleep state.

30. The system of claim 31 wherein the throttling emulator causing the processor to load the SMI timer causes the processor to:

load the SMI timer in a chipset.